UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/622,022	07/17/2003	John B. Bley	WILY-01016US0	1684

28554 7590 11/14/2007

VIERRA MAGEN MARCUS & DENIRO LLP
575 MARKET STREET SUITE 2500
SAN FRANCISCO, CA 94105

EXAMINER	
CHOW, CHIH CHING	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
11/14/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/622,022	BLEY ET AL.	
	Examiner	Art Unit	
	Chih-Ching Chow	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 September 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-47* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 July 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to amendment dated September 11, 2007.

2. Per Applicants' request, independent claims 1, 29, and 36 have been amended.

3. Claims 1-47 remain pending.

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on September 11, 2007 has been entered.

Response to Arguments

5. Applicant's arguments with respect to independent claims 1, 29, and 36 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-6, 8-9, 12-18, 20-22, 25-27, 29-34, 36-39, 41-46 are rejected under 35 U.S.C. 102(b) as being anticipated by US Patent No. 6,131,187, by Chow et al., hereinafter "Chow".

As Per claim 1, Chow discloses:

- ***A method for adding functionality in order to access information, comprising: accessing existing object code, said existing object code includes a first method, said first method ~~is capable of providing~~ <u>produces</u> a result when***

said first method is executed; and adding new code to said first method, said new code provides said result to an additional method.

Chow's disclosure teaches a method to access existing object code, see Chow's Fig. 2, item 42, 'step through bytecode within a bytecode stream', and see Chow's column 3, line 65 into column 4, line 7, "An exception is a software error condition that **interrupts the flow of a bytecode program**. Within a **Java™** virtual machine, an exception always occurs when the **bytecode program executes** a **throw instruction** (*produces a result when said first method is executed).* The throw instruction passes control to an associated catch block. A catch block is a section of code designed to process the thrown exception. A catch block must immediately follow a **try block** or another **catch block**. The try block encloses a sequence of statements from which a throw can originate." – the original bytecode program is the first method. Further lines 40-67, "The **pertinent information** preferably includes a program counter, a stack pointer, an exception handler program counter (if present), the catch type for the exception handler (in necessary), other information such as branch address, and various flags. All pertinent information in each entry 36 of bytecode information array 35 is obtained by **stepping through each bytecode** within a given bytecode stream (executing the first method), as depicted in block 42." And the pseudocode Procedure StepThruCode (access existing object code). Further, see pseudocode in column 5, lines 10-35, wherein the code (try and catch) lead to the exception handler is the new code, the new code provides results (the exception) to the additional method (exception handler), which is the procedures within the exception handler; the additional method is for tracing and monitoring purposes, such as building up the bytecode information array (*adding new code to said first method, said new code provides said result to an additional method).*

As Per claim 2, Chow discloses:

- ***The method according to claim 1, wherein:***

  Said result includes a data item to be returned by said first method.

Claim 1 rejection is incorporated, further see Chow's column 4, lines 40-66, the 'pc', 'sp', 'eh', ...etc. are all data items returned by executing the procedure (the first method).

As Per claim 3, Chow discloses:

- ***A method according to claim 1, wherein: said result includes a reference to an exception.***

Claim 1 rejection is incorporated, further see Chow's column 4, lines 2-4, "the throw instruction passes control to an associated catch block. A catch block is a section of code designed to **process the thrown exception**." – the result includes a reference to an exception.

As Per claim 4, Chow discloses:

- ***A method according to claim 1, wherein said step of adding new code includes: adding code that stores said result for said first method from an operand stack; adding code that prepares said operand stack for an invocation of said additional method; adding code that invokes said additional method, including providing said result to said additional method; and adding code that resets said operand stack with respect to said result to a state existing prior to storing said result.***

Claim 1 rejection is incorporated, further see Chow's column 3, lines 42-46, "During **normal execution of the bytecode program** by bytecode program interpreter 31, bytecode program interpreter 31 must continually **monitor the operand stack** for overflows (i.e., attempting to **add more data to the stack**

than the stack can store) and underflows (i.e., attempting to pop data off the stack when the stack is empty)." -- *operand stack for an invocation.* Also see Chow's column 5, lines 63-66, "If no exception occurred for an instruction working on the same operand(s), then the **generation of exception handling code**, such as Null PointerException and ArrayOutOfBoundException, for all the subsequent instructions of the same opcode and operands is not required." -- *adding code that resets said operand stack.*

As Per claim 5, Chow discloses:

- ***A method according to claim 4, wherein said step of adding new code further includes: adding code that returns said result after resetting said operand stack, said result is a return value.***

See claim 4 rejection, further see Chow's column 3, lines 29-36, "Bytecode program verifier 30 is an executable program that verifies operand data type compatibility and **stack manipulations** properness in a specified bytecode program prior to the execution of the bytecode program by processor 22 under bytecode program interpreter 31. Each bytecode program has an associated **verification status value that is initially set** when the bytecode program is being downloaded from another location, such as a file server." *(reset operand stack).*

As Per claim 6, Chow discloses:

- ***A method according to claim 4, wherein: said result includes an exception; and said step of adding new code further includes adding code that throws said exception after said step of resetting, said result represents an exception.***

Claim 4 rejection is incorporated, for rest of claim 6 feature see claim 1 rejection.

As Per claim 8, Chow discloses:

- ***A method according to claim 1, wherein: said first method is a ~~Java~~ JAVA method.***

Claim 4 rejection is incorporated, for JAVA feature see Chow's column 3, line 65 into column 4, line 7, "An exception is a software error condition that **interrupts the flow of a bytecode program**. Within a **Java™** virtual machine, an exception always occurs when the **bytecode program executes a throw instruction** (*produces a result when said first method is executed).* The throw instruction passes control to an associated catch block. A catch block is a section of code designed to process the thrown exception", and column 5, lines 35-39, "As has been described, the present invention provides an improved method for translating exception handling semantics of a bytecode class file within a computer system. Instead of following the exception **handling mechanism of a Java™ class file**, the present invention analyzes the **exception handling frames** defined in the class file."

As Per claim 9, Chow discloses:

- ***A method according to claim 1, wherein said step of adding new code includes: adding start byte code; adjusting byte code indices; adding exit byte code; and modifying an exception table for said first method.***

Claim 1 rejection is incorporated, further for of claim 9 feature see Chow's column 1, lines 52-61, "The exception handling semantics, embedded within an exception handling structure (or exception framelist) of a bytecode class file, are designed to allow the bytecode interpreter to **handle exceptions that occur** during an enclosed try statement. However, the exception handling structure prevents the translated high-level code from being optimized by the compiler because the artificial exception ranges **kept in the exception table** of a method

preclude the final natively compiled instructions from being rescheduled for optimal performance." And Chow's column 4, Table I contains information similar to the start byte code, adjusting byte code, indices, counts, for exception handlers.

As Per claim 12, Chow discloses:

- ***A method to provide access information, comprising: storing a result for a first method from an operand stack; preparing said operand stack for an invocation of a second method; invoking said second method, including providing said result to said second method; and resetting said operand stack with respect to said result to a state existing prior to said step of storing said result.***

Chow's disclosure is for accessing stored information see Chow's Fig. 1, and description in column 3, lines 42-47, "bytecode program interpreter 31 must continually monitor the **operand stack** for overflows (i.e., attempting to **add more data to the stack than the stack can store**) and underflows (i.e., attempting to pop data off the stack when the stack is empty). Such stack monitoring must normally be performed for all bytecodes that change the status of the stack." -- results are stored in an operand stack.

As Per claim 13, Chow discloses:

- ***A method according to claim 12, wherein: said result includes a data item to be returned by said first method.***

Claim 12 rejection is incorporated, further see claim 2 rejection.

As Per claim 14, Chow discloses:

- ***A method according to claim 13, further comprising: returning said result after said step of resetting.***

Claim 13 rejection is incorporated, further see claim 5 rejection.

As Per claim 15, Chow discloses:

- ***A method according to claim 12, wherein: said result includes a reference to an exception.***

Claim 12 rejection is incorporated, further see claim 3 rejection.

As Per claim 16, Cow discloses:

- ***A method according to claim 15, further comprising: throwing said exception after said step of resetting.***

For claim 15 feature see claim 15 rejection, for rest of claim 16 feature see claim 5 rejection.

As Per claim 17, Chow discloses:

- ***A method according to claim 12, further comprising: performing said second method in response to said step of invoking.***

For claim 12 feature see claim 12 rejection, for rest of claim 17 feature see claim 7 rejection.

As Per claim 18, Chow discloses:

- ***A method according to claim 12, further comprising: performing one or more instructions of said first method prior to said step of storing said result, said step of performing one or more instructions includes generating said result.***

For claim 12 feature see claim 12 rejection, for rest of claim 18 feature see

claim 1 rejection.

As Per claim 20, Chow discloses:

- ***A method according to claim 12, wherein: said first method is a ~~Java~~ JAVA method.***

For claim 12 feature see claim 12 rejection, for rest of claim 20 feature see claim 8 rejection.

As Per claim 21, Chow discloses:

- ***A method according to claim 12, further comprising: modifying byte code for said first method to add new code that performs said steps of storing, preparing, invoking and resetting.***

For claim 12 feature see claim 12 rejection, for rest of claim 21 feature see claim 9 rejection.

As Per claim 22, Chow discloses:

- ***A method according to claim 21, wherein said step of modifying includes: adding start byte code; adjusting byte code indices; adding exit byte code; and modifying an exception table for said first method.***

For claim 21 feature see claim 21 rejection, for rest of claim 22 feature see claim 9 rejection.

As Per claim 25, Chow discloses:

- ***A method according to claim 12, further comprising: performing said second method, including accessing said result.***

Claim 12 rejection is incorporated, further see claims 1, 4 rejections.

As Per claim 26, Chow discloses:

- ***A method according to claim 12, further comprising: performing said second method, including storing said result for use outside of a thread that includes said first method.***

Claim 12 rejection is incorporated, further see claim 1 rejection.

As Per claim 27, Chow discloses:

- ***A method according to claim 12, further comprising: performing said second method in response to said step of invoking, said second method stores said result; performing one or more instructions of said first method prior to said step of storing said result, said step of performing one or more instructions includes generating said result, said first method is a ~~Java~~ JAVA method; and modifying byte code for said first method to add new code that performs said steps of storing, preparing, invoking and resetting.***

For claim 12 feature see claim 12 rejection, for rest of claim 27 feature see claims 1, and 8 rejections.

As Per claim 29, Chow discloses:

- ***One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising: accessing existing object code, said existing object code includes a first method, said first method ~~is capable of providing~~ produces a result when said first method is executed; and adding new code to said first method, said new code provides said result to an additional method.***

Chow's teaching also applies for one or more processor readable storage devices having processor readable code; claim 29 is one or more processor

readable storage devices' version of claim 1, it is rejected on the same basis as claim 1.

As Per claim 30, Chow discloses:

- ***One or more processor readable storage devices according to claim 29, wherein: said result is a data item to be returned by said first method.***

Claim 29 rejection is incorporated, further see claim 2 rejection.

As Per claim 31, Chow discloses:

- ***One or more processor readable storage devices according to claim 29, wherein: said result is a reference to an exception.***

Claim 29 rejection is incorporated, further see claim 3 rejection.

As Per claim 32, Chow discloses:

- ***One or more processor readable storage devices according to claim 29, wherein said step of adding new code includes: adding code that stores said result for said first method from an operand stack; adding code that prepares said operand stack for an invocation of said additional method; adding code that invokes said additional method, including providing said result to said additional method; and adding code that resets said operand stack with respect to said result to a state existing prior to storing said result.***

For claim 29 feature see claim 29 rejection, for rest of claim 32 feature see claim 4 rejection.

As Per claim 33, Chow discloses:

- ***One or more processor readable storage devices according to claim 29, wherein: said first method is a ~~Java~~ JAVA method.***

For claim 29 feature see claim 29 rejection, for rest of claim 33 feature see claim 8 rejection.

As Per claim 34, Chow discloses:

- ***One or more processor readable storage devices according to claim 29, wherein said step of adding new code includes: adding start byte code; adjusting byte code indices; adding exit byte code; and modifying an exception table for said first-method.***

Claim 29 rejection is incorporated, further see claim 9 rejection.

As Per claim 36, Chow discloses:

- ***An apparatus that adds functionality in order to access information, comprising: a communication interface; a processor readable storage device; and one or more processors in communication with said processor readable storage device and said communication interface, said one or more processors perform a method comprising: access existing object code, said existing object code includes a first method, said first method ~~is capable of providing~~ produces a result when said first method is executed, and adding new code to said first method, said new code provides said result ~~value~~ to an additional method.***

Chow's teaching also applies for an apparatus, which comprising a communication interface (see Chow's Fig. 1) and a processor readable storage device (see Chow's Fig. 1), and one or more processors in communication with the communication interface; claim 36 is an apparatus version of claim 1, it is rejected on the same basis as claim 1.

As Per claim 37, Chow discloses:

- ***An apparatus according to claim 36, wherein: said result is a data item or a reference to an exception.***

Claim 36 rejection is incorporated, further see claim 3 rejection.

As Per claim 38, Chow discloses:

- ***An apparatus according to claim 36, wherein said step of adding new code includes: adding code that stores said result for said first method from an operand stack; adding code that prepares said operand stack for an invocation of said additional method; adding code that invokes said additional method, including providing said result to said additional method; and adding code that resets said operand stack with respect to said result to a state existing prior to storing said result.***

Claim 36 rejection is incorporated, further see claim 4 rejection.

As Per claim 39, Nilsson discloses:

- ***An apparatus according to claim 36, wherein said step of adding new code includes: adding start ~~Java~~ JAVA byte code; adjusting ~~Java~~ JAVA byte code indices; adding exit ~~Java~~ JAVA byte code; and modifying an exception table for said first method.***

For claim 36 feature see claim 36 rejection, for rest of claim 39 feature see claim 8 and claim 9 rejection.

As Per claim 41, Chow discloses:

- ***An apparatus that adds functionality to existing code in order to access information, comprising: a communication interface; a processor readable storage device; and one or more processors in communication with said***

processor readable storage device and said communication interface, said one or more processors perform a method comprising: storing a result for a first method from an operand stack, preparing said operand stack for an invocation of a second method, invoking said second method, including providing said result to said second method, and resetting said operand stack with respect to said result to a state existing prior to said step of storing said result.

Claim 41 is an apparatus version of claim 1, it is rejected on the same basis as claims, 1, 4, 36 rejections.

As Per claim 42, Chow discloses:

- ***An apparatus according to claim 41, wherein: said result is a data item to be returned by said first method.***

Claim 41 rejection is incorporated, further see claim 2 rejection.

As Per claim 43, Chow discloses:

- ***An apparatus according to claim 41, wherein: said result is a reference to an exception.***

Claim 41 rejection is incorporated, further see claim 3 rejection.

As Per claim 44, Nilsson discloses:

- ***An apparatus according to claim 41, wherein: said first method is a ~~Java~~ JAVA method.***

For claim 41 feature see claim 41 rejection, for rest of claim 44 feature see claim 8 rejection.

As Per claim 45, Chow discloses:

- ***An apparatus according to claim 41, wherein said method further comprises: modifying byte code for said first method to add new code that performs said steps of storing, preparing, invoking and resetting.***

Claim 41 rejection is incorporated, further see claim 9 rejection.

As Per claim 46, Chow discloses:

- ***An apparatus according to claim 45, wherein said step of modifying includes the steps of: adding start byte code; adjusting byte code indices; adding exit byte code; and modifying an exception table for said first method.***

Claim 45 rejection is incorporated, further see claim 9 rejection.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 7, 10, 11, 19, 23, 24, 28, 35, 40, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 6,131,187 by Chow et al., hereinafter "Chow", in view of US Patent No. 5, 628,016 by Kukol, hereinafter "kukol".

As Per claim 7,

- ***A method according to claim 4, wherein said step of adding new code further includes: adding code that jumps to a subroutine representing a Finally block after invoking said additional method; and adding code that is to be executed after returning from said subroutine.***

Claim 4 rejection is incorporated, Chow teaches all aspects of claim 7, but he does not disclose 'Finally block' explicitly, however, Kukol teaches in an analogous prior art, see Kukol's Fig. 9, which illustrating occurrence of a **try/finally block** within a setjmp block."

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Chow's disclosure of the exception handler, with try and catch blocks by the 'finally block' taught by Kukol. The modification would be obvious because one of ordinary skill in the art would be motivated by the finalization code is executed (See Kukol's column 24, lines 50-58).

As Per claim 10,

- ***A method according to claim 9, wherein said step of adding exit byte code includes: adding byte code to report said result and jump to a subroutine representing a Finally block; adding byte code to report an exception and jump to said subroutine representing said Finally block; and adding byte code for said subroutine representing said Finally block.***

Claim 9 rejection is incorporated, Chow teaches all aspects of claim 10, but he does not disclose 'Finally block' explicitly, however, Kukol teaches in an analogous prior art, see Kukol's Fig. 9, which illustrating occurrence of a **try/finally block** within a setjmp block."

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Chow's disclosure of the exception handler, with try and catch blocks by the 'finally block' taught by Kukol. The modification would be obvious because one of ordinary skill in the art would be motivated by the finalization code is executed (See Kukol's column 24, lines 50-58).

As Per claim 11,

- ***A method according to claim 1, wherein said step of adding new code includes: adding Try-Finally functionality.***

Claim 1 rejection is incorporated, Chow teaches all aspects of claim 10, but he does not disclose 'Finally block' explicitly, however, Kukol teaches in an analogous prior art, see Kukol's Fig. 9, which illustrating occurrence of a **try/finally block** within a setjmp block."

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Chow's disclosure of the exception handler, with try and catch blocks by the 'finally block' taught by Kukol. The modification would be obvious because one of ordinary skill in the art would be motivated by the finalization code is executed (See Kukol's column 24, lines 50-58).

As Per claim 19,

- ***A method according to claim 12, further comprising: returning said result subsequent to said step of resetting; jumping to a subroutine representing a Finally block after invoking said second method and prior to returning said result; and returning from said subroutine prior to returning said result.***

For claim 12 feature see claim 12 rejection, for rest of claim 19 feature see claim 7 rejection.

As Per claim 23,

- ***A method according to claim 22, wherein said step of adding exit bytes code includes: adding byte code to report said result and jump to a subroutine representing a Finally block; adding byte code to report an, exception and jump to said subroutine representing said Finally block; and adding byte code for said subroutine representing said Finally block.***

Claim 22 rejection is incorporated, further for see claim 10 rejection.

As Per claim 24,

- ***A method according to claim 21, wherein said step of modifying includes: adding Try-Finally functionality.***

Claim 21 rejection is incorporated, further for see claim 11 rejection.

As Per claim 28,

- ***A method according to claim 27, further comprising: returning said result; jumping to a subroutine representing a Finally block after invoking said second method and prior to returning said result; and returning from said subroutine prior to returning said result.***

For claim 27 feature see claim 27 rejection, for rest of claim 28 feature see claim 10 rejection.

As Per claim 35,

- ***One or more processor readable storage devices according to claim 34, wherein said step of adding exit byte code includes: adding byte code to report said result and jump to a subroutine representing a Finally block; adding byte code to report an exception and jump to said subroutine representing said Finally block; and adding byte code for said subroutine representing said Finally block.***

Claim 34 rejection is incorporated, further see claim 10 rejection.

As Per claim 40,

- ***An apparatus according to claim 39, wherein said step of adding exit byte code includes: adding byte code to report said result and jump to a subroutine***

representing a Finally block; adding byte code to report an exception and jump to said subroutine representing said Finally block; and adding byte code for said subroutine representing said Finally block.

For claim 39 feature see claim 39 rejection, for rest of claim 40 feature see claim 7 rejection.

As Per claim 47,

- ***An apparatus according to claim 46, wherein said step of adding exit byte code includes: adding byte code to report said result and jump to a subroutine representing a Finally block; adding byte code to report an exception and jump to said subroutine representing said Finally block; and adding byte code for said subroutine representing said Finally block.***

Claim 46 rejection is incorporated, further see claim 10 rejection.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Kramskoy et al. US 2002/0112227, discloses a dynamic compiler and method of compiling code to generate a dominate path and handle exceptions. The dynamic compiler includes an execution history recorder that is configured to record the number of times a fragment of code is interpreted.

Marshed et al., US Patent No. 6,760,903, discloses a method of execution control is intercepted at a first communication point between a calling function and a called function for a cross execution context call. The calling function is associated with a first execution context and the called function is associated with a second execution context. The call origin information is extracted prior to executing the called function at a second communication point.

Crank et al., US Patent No. 5,583,988, discloses a method and apparatus for performing runtime checking during program execution in a compiled environment using the full ANSI-C programming language. The present invention detects a number of errors during runtime that cannot be found by a compiler at the precise moment that a respective C language restriction is violated. The present invention also provides the user with a direct indication of the problem, thus saving debugging time.

Yates et al., US Patent No. 7,065,633, discloses a computer concurrently executes a first operating system coded in a RISC instruction set and a second operating system coded in a CISC instruction set. When an exception is raised while executing a program coded in the RISC instruction set, an execution thread may be initiated under the CISC operating system. The exception may be delivered to the initiated thread for handling by the CISC operating system.

Draine et al., US 20040250175, discloses an error/exception helper may provide tailored help when an error such as an exception is generated. A source program editor interface may be displayed and/or focus given to the program editor interface. An error/exception bubble or tool tip may be displayed, which, in one embodiment of the invention, points to the line of code that generated the exception. The error/exception bubble may include a link to a help topic or the actual help text may be displayed within the bubble.

11. The following summarizes the status of the claims:

35 USC § 102 rejection: Claims 1-6, 8, 9, 12-18, 20-22, 25-27, 29-34, 36-39, 41-46.

35 USC § 103 rejection: Claims 7, 10, 11, 19, 23, 24, 28, 35, 40, and 47

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature of relating to the status of this application should be directed to the **TC2100 Group receptionist: 571-272-2100**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chih-Ching Chow
Examiner
Art Unit 2191
November 8, 2007

CC

MARY STEELMAN
PRIMARY EXAMINER